A New Interactive Hashing Theorem*

Iftach Haitner† Omer Reingold†

April 13, 2008

Abstract

Interactive hashing, introduced by Naor, Ostrovsky, Venkatesan and Yung (CRYPTO '92), plays an important role in many cryptographic protocols. In particular, it is a major component in all known constructions of statistically hiding and computationally binding commitment schemes and of zero-knowledge arguments based on general one-way permutations and on one-way functions. Interactive hashing with respect to a one-way permutation f, is a two-party protocol that enables a sender that knows $y = f(x)$ to transfer a random hash $z = h(y)$ to a receiver. The receiver is guaranteed that the sender is committed to y (in the sense that it cannot come up with x and x' such that $f(x) \neq f(x')$, but $h(f(x)) = h(f(x')) = z$). The sender is guaranteed that the receiver does not learn any additional information on y. In particular, when h is a two-to-one hash function, the receiver does not learn which of the two preimages $\{y, y'\} = h^{-1}(z)$ is the one the sender can invert with respect to f.

This paper reexamines the notion of interactive hashing. We give an alternative proof for the Naor et al. protocol, which seems to us significantly simpler and more intuitive than the original one. Moreover, the new proof achieves much better parameters (in terms of how security preserving the reduction is). Finally, our proof implies a more versatile interactive hashing theorem for a more general setting than that of the Naor et al. protocol. One generalization relates to the selection of hash function h (allowing much more flexibility). More importantly, the theorem applies to the case where the underlying function f is hard-to-invert only on some given (possibly sparse) subset of the output strings. In other words, the theorem is tuned towards hashing of a value y that may be distributed over a sparse subset of the domain (rather than uniform on the entire domain as a random output of a one-way permutation is).

Our interest in interactive hashing is in part as a very appealing object (i.e., independent of any particular application). Furthermore, a major motivation for looking into interactive hashing is towards improving and simplifying previous constructions of statistical zero-knowledge and statistical commitments (that employ interactive hashing as a central building block). We make some preliminary progress in this direction as well.

Keywords: cryptography, interactive hashing, statistically hiding and computationally binding commitments.

*Preliminary version of this paper appeared as [HR07].

†Weizmann Institute of Science, Rehovot, Israel. E-mail: {iftach.haitner,omer.reingold}@weizmann.ac.il.
Research supported by grant no. 1300/05 from the Israel Science Foundation.

1 Introduction

Interactive hashing, introduced by Naor, Ostrovsky, Venkatesan and Yung [NOVY98], is a protocol that allows a sender S to commit to a particular value while only reviling to a receiver R some predefined information of this value. More specifically, S commits to a value y while only reviling to R the value $(h, h(y))$, where h is some random hash function (we defer additional details on the choice of hash function). The two security properties of interactive hashing are *binding* (namely, S is bounded by the protocol to at most one value of y) and *hiding* (namely, R does not learn any impermissible information about y). As in [NOVY98], we will consider in this work interactive hashing where the hiding property is statistical (i.e., the protocol preserves the secrecy of y even against an all-powerful R), and the binding property is computational (i.e., it assumes that S is computationally bounded).

Interactive hashing (in the flavor mentioned above) is closely related and to a large extent motivated by the fundamental notion of statistically hiding commitments (i.e., statistically hiding and computationally binding commitment schemes). In statistically hiding commitments, we again have a protocol between a sender S and a receiver R. However, here the sender S commits to y without reviling *any* information about y. Statistically hiding commitments can be used as a building block in constructions of statistical zero-knowledge arguments [BCC88, NOVY98] or certain coin-tossing protocols [Lin03]. More generally, they have the following advantage over computationally hiding commitment schemes when used within protocols in which certain commitments are never revealed: in such a scenario, it need only be infeasible to violate the binding property *during the period of time the protocol is run*, whereas the committed values will remain hidden *forever* (i.e., regardless of how much time the receiver invests after completion of the protocol).

The relation between interactive hashing and statistically hiding commitments goes beyond the similarity in definitions. On one hand, interactive hashing can easily be implemented using commitment schemes (simply commit to y using the commitment scheme and reveal whatever information needed on y in the clear). On the other hand, one of the main applications of interactive hashing protocols is for constructing statistically hiding commitment schemes. Indeed, interactive hashing is a major component in all known constructions (listed below) of statistically hiding commitment that are based on, possibly restricted types of, one-way functions.

Naor et al. [NOVY98] use their interactive hashing protocol (from now on the NOVY protocol) in order to construct statistically hiding commitments based on any one-way permutation. Haitner et al. [HHK+05] make progress by using the NOVY protocol to construct statistically hiding commitments based on regular one-way functions and also on the so called approximable-size one-way functions. Recently, Haitner et al. [HNO+07] constructed statistically hiding commitments based on *any* one-way functions. Not surprisingly, interactive hashing is heavily used in the their construction.

Interactive hashing is also used by several other cryptographic protocols [OVY93a, OVY92, OVY93b]. In addition, it is used in "information theoretic setting" (i.e., no hardness assumptions are assumed) such as [CCM98, CS06, DHRS04, NV06].

A possible drawback of [HNO+07] is that the construction is rather inefficient and complex. Indeed, a major motivation for looking into interactive hashing is to simplify constructions of statistically hiding commitment schemes based on any one-way functions. [1] However, before

[1] Actually, [HNO+07] uses one of the results given in this work (Theorem 5.2). This is unlike [NOV06], which is part of the two papers that [HNO+07] combines, where an independent interactive hashing theorem is given. Yet, it

discussing our results and their applications let us have a closer look into the notion of interactive hashing.

1.1 Interactive Hashing in the Setting of One-Way Permutations

Consider the following two-party protocol between a sender S and a receiver R: The sender chooses a random element $x \in \{0,1\}^n$ and sets $y = f(x)$, where $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ is a one-way permutation. Next, the receiver selects a random two-to-one hash function $h\colon \{0,1\}^n \mapsto \{0,1\}^{n-1}$ and sends its description to S. Finally, S sends $z = h(y)$ back to R. Note that if both parties follow the protocol, then the following "binding" property is guaranteed: It is not feasible for S to find a *second* element $x' \in \{0,1\}^n$ such that $f(x') \neq f(x)$ but $h(f(x')) = h(f(x)) = z$, although (exactly one) such element x' does exist. The reason is that the task of finding such x' can easily be shown to be equivalent in hardness to inverting f on a random output element (the latter task is assumed to be hard by the one-wayness of f). Furthermore, we are guaranteed to have the following "hiding" property: Let y_1 and y_2 be the two preimages of z w.r.t. h. Given R's view of the communication (i.e., given the values h and z), it is indistinguishable whether the random element chosen by S is $x_1 = f^{-1}(y_1)$ or $x_2 = f^{-1}(y_2)$. In this sense, S has committed to a bit (which indicates if it can produce the inverse of y_1 or that of y_2). This bit is statistically hidden from R.

What happens, however, if S selects x only *after* seeing h? In such a case, it is quite plausible that S would be able to "cheat" by producing $x, x' \in \{0,1\}^n$ such that $f(x) \neq f(x')$ but $h(f(x')) = h(f(x)) = z$. [2] The NOVY interactive hashing protocol prevents exactly such cheating. For that it employs a specific family of hash functions such that each one of its functions h can be decomposed into $n-1$ Boolean functions $h_1, \ldots, h_{n-1}$, where $h(x) = h_1(x), \ldots, h_{n-1}(x)$. [3] In the NOVY protocol, instead of sending h at once as described above, R sends a single Boolean function h_i in each round. In return, the honest sender sends a bit $z_i = h_i(f(x))$. What about a cheating sender? Intuitively, a cheating sender has a significantly smaller leeway for cheating as it can no longer wait in selecting x till it receives the entire description of h. Still, it is highly non-trivial to argue (formally or even intuitively) that restricting the sender by adding interaction in this manner is sufficient in order to prevent the sender from cheating. Perhaps surprisingly, Naor et al. [NOVY98] have shown that their protocol has the binding property even against a cheating sender (namely, even a cheating sender cannot produce $x, x' \in \{0,1\}^n$ such that $f(x) \neq f(x')$, but $h(f(x')) = h(f(x)) = z$).

1.2 Interactive Hashing in the Sparse Case

The NOVY interactive hashing protocol applies to one-way permutations and easily implies the existence of statistically hiding commitments from any one-way permutation. [4] How about constructing statistically hiding commitments from, say, regular one-way functions (one-way functions where every output value has the same number of preimages)? In such a case we would like to

seems that [HNO+07] does not take a full advantage of the results given in this chapter.

[2] Assume for example that the one-way permutation equals the identity function on the set T of all strings that start with $n/4$ zeros (where n is the input length). Now given a hash function h all the cheating sender has to do is to find a collision $y_1 \neq y_2$, where $y_1, y_2 \in T$, such that $h(y_1) = h(y_2)$. Such a collision is likely to exist by the birthday paradox, and for many families of hash functions finding such a collision is easy.

[3] For more details on the definition of this family of hash functions see Section 5.

[4] Actually, the NOVY commitment schemes are even stronger being perfectly hiding.

interactively hash a value y (a random output of the one-way function) which is uniformly distributed in some subset L of $\{0,1\}^n$ (rather than uniformly distributed in all of $\{0,1\}^n$ as in the case of one-way permutations). What is the difficulty in directly hashing a value y that is taken from a set L that is sparse in $\{0,1\}^n$? The NOVY-theorem guarantees that when hashing y with $h\colon \{0,1\}^n \mapsto \{0,1\}^{n-1}$ the sender is committed to a single value y (as shown in [NOV06] this holds even if the output of h is a bit shorter). However, when h outputs so many bits then most likely $h(y)$ completely determines y and statistical hiding is lost.

Facing the aforementioned difficulty, Haitner et al. [HHK+05] first make the following observation: the NOVY protocol is still meaningful even when hashing a value y which is taken from a distribution that is "dense" in $\{0,1\}^n$ (a bit more formally we would like the distribution to be sufficiently close to having min-entropy $n - O(\log n)$). In particular, if the one-way function is length-preserving poly-to-one (i.e., each output has at most polynomial number of preimages in the image set of f), then the NOVY-protocol can be applied as is to give some weak form of statistically hiding commitments that can later be amplified to full-fledge statistically hiding commitments. To handle any regular one-way function, [HHK+05] applies additional layer of (non interactive) hashing to reduce to the dense case. This implies a construction of statistically hiding commitments from any regular one-way function with known image size. Interactive hashing in the sparse case arises in other works as well, most notably in the construction of statistical zero-knowledge arguments from any one-way function [NOV06].

1.3 Our Results

We introduce an alternative proof for the NOVY protocol that relies in parts on the original proof due to [NOVY98] (the NOVY proof), but still seems to us significantly simpler. The proof follows a simple intuition that is sketched below in this section. Moreover, the parameters achieved by our proof are an improvement compared with the original ones. Given an algorithm A that breaks the binding property with probability ε_A, we get an algorithm that inverts the one-way permutation in comparable time and with inverting probability $\varepsilon_A^2 \cdot \mathrm{poly}(n)$ (where n is the hash function input length). This is a substantial improvement and is much closer to natural limitations of the proof technique (see discussion in Section 6). [5]

In addition to being simpler and more security preserving, the new proof implies a more general interactive hashing theorem. The new theorem applies to every family of hash functions that is a product of Boolean families of pairwise independent hash functions (and not only to the special family of two-to-one hash functions used by [NOVY98, NOV06]). More importantly, the new theorem directly applies to the "sparse case". Let $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ be an efficiently computable function and let $L \subseteq \{0,1\}^{\ell(n)}$. As mentioned above, when hashing a value $y \in L$, the NOVY proof only promises binding when using a hash function that outputs almost n bits. in such a case, however, y is likely to be completely determined by $h(y)$ and statistical hiding cannot be guaranteed. Our theorem applies even when hashing to roughly $m = \lfloor \log(|L|) \rfloor$ bits. In particular, when h is taken from a family of hash functions $\mathcal{H}^m\colon \{0,1\}^{\ell(n)} \mapsto \{0,1\}^m$ that is a product of m families of pairwise independent Boolean hash functions, we can show that a close variant of the NOVY protocol possesses the following binding property: If f is hard to invert on the uniform distribution over L, then no polynomially-bounded sender S^* (even one which arbitrarily

[5] We note that independently of our work, [NOV06] recently presented an $\varepsilon_A^3 \cdot \mathrm{poly}(n)$ reduction. See discussion below for more detail on their work.

deviates from the protocol) can find two elements $x, x' \in f^{-1}(L)$ such that $f(x) \neq f(x')$ but $h(f(x)) = h(f(x')) = z$ (where z is the value determined by the protocol as $h(y)$).

Finally, we consider an interactive hashing protocol that uses general pairwise independent hash functions (and not necessarily Boolean), and prove the binding of this protocol holds if the underlying one-way function is hard on the average against algorithm that runs in time $\text{poly}(n)2^s$ time, where s is the hash functions' output length. The round complexity of our protocol, for any hardness of the underlying one-way function, matches the recent lower bound of Haitner et al. [HHRS07].

1.4 Applications of the New Theorem

As an easy corollary, we use the new theorem to derive a direct construction of statistically hiding commitment based on known regular one-way functions (and thus reprove [HHK+05]). We also believe that our new result can also used to simply the construction of two-phase commitment schemes given in [HNO+07] and thus to simplify their construction of statistically hiding commitments from any one-way functions.

1.5 Related Work

We note that independently of our work, Nguyen et al. [NOV06] give a new proof for the NOVY protocol. Their proof follows the proof of [NOVY98] more closely than ours but still introduces various simplifications and parameter improvements. The main goal of the new proof is to generalize the protocol such that it allows hashing with a hash function that is poly-to-one rather than two-to-one as in [NOVY98]. In other words, they analyze the NOVY protocol with $n - O(\log n)$ rounds rather than $n - 1$ rounds in [NOVY98]. For a comparison between the parameters obtained by [NOVY98], [NOV06] and this work, see Remark 3.10.

1.6 Relations

Following [NOV06], we state our protocol, and proof, in the more general setting of binary relations rather than functions. For instance, given a binary relation W that is hard to satisfy (i.e., given y it is hard to find x such that $(x, y) \in W$), we prove that following our interactive hashing protocol, S cannot find two pairs $(x_0, y_0), (x_1, y_1) \in W$ such that both y_0 and y_1 are consistent with the protocol, but $y_0 \neq y_1$. Note that since every function, f, defines the natural binary relation $(x, f(x))$, any result w.r.t. binary relations implies an equivalent result w.r.t. functions.

1.7 Cheating Receiver

In our new interactive hashing theorem, the hiding property of a cheating receiver is specified only with respect to a semi-honest receiver (a.k.a. honest-but-curious). A malicious receiver can learn $h(y)$ where h is not necessarily uniformly distributed. In fact, h can be determined adaptively based on partial knowledge of $h(y)$ (specifically, h_i is selected after learning the first $i - 1$ bits of $h(y)$). The NOVY protocol provides a cheating receiver exactly the same power in selecting h. Nevertheless, when y is selected uniformly in $\{0, 1\}^n$ and h is two-to-one, then regardless of how the receiver selects h there is one bit of knowledge on y that remains completely hidden from the receiver. That is, given $z = h(y)$ there are two possibilities to the value of y, the hidden bit specifies which of these two values is the right one. In the general case, when y is distributed over a sparse

subset, one should take more care in estimating the power of a cheating receiver. We note that assuming the existence of one-way functions, in various settings, one can assume without loss of generality that the receiver is semi-honest. In particular, this is the case for statistically hiding commitments ([HHK+05]).

1.8 Proof Idea

We discuss our binding proof in the most basic setting where $f\colon \{0,1\}^n \mapsto \{0,1\}^n$ is a one-way permutation and $L = \{0,1\}^n$ (the proof of general case is not significantly different). Our protocol consists of $m = n - O(\log(n))$ rounds. First, S selects a random element $x \in \{0,1\}^n$, then in each round, R selects a random Boolean pairwise-independent hash function h_i and S replies with $z_i = h_i(f(x))$.

Let A be an algorithm that plays the sender's role in the protocol and at the end of the protocol outputs two elements $x_1, x_2 \in \{0,1\}^n$. Assume that with some noticeable probability ε, it holds that $f(x_1) \neq f(x_2) \in \mathsf{Consist}(h_1, \ldots, h_m)$, where $\mathsf{Consist}(h_1, \ldots, h_m) = \{y \in \{0,1\}^n \colon \forall i \in [m] \;\; h_i(y) = z_i\}$. It is easy to use A in order to construct an algorithm that inverts f with probability $\frac{\varepsilon}{2^n}$. Given input y, the algorithm chooses the hash functions at random and returns one of the two values that A outputs. [6]

Let's imagine that instead we are trying to invert f on the following distribution. The first $k = m - \log(\frac{1}{\varepsilon}) - c \cdot \log(n)$ (for some constant $c \in \mathbb{N}$ determined by the analysis) Boolean hash functions, $h_1, \ldots, h_k$, are chosen at random and only then a random element, y, is uniformly drawn from $\mathsf{Consist}(h_1, \ldots, h_k)$. We call the distribution induced on $(y, h_1, \ldots, h_k)$ by the above process D_{Uni}. On the average, A has probability ε to cheat even when conditioned on $h_1, \ldots, h_k$ being selected. Also note that by the pairwise independence of the h's, the size of $\mathsf{Consist}(h_1, \ldots, h_k)$ is, with high enough probability, about $\frac{2^n}{2^{m-k}} = \frac{n^c}{\varepsilon}$. It follows that the naive algorithm, which selects the rest of the hash functions at random and returns one of A's answers, inverts f with probability close to $\frac{\varepsilon^2}{n^c}$ over D_{Uni}.

Let's try to emulate the above setting on a random $y \in \{0,1\}^n$. Namely, given a uniformly chosen $y \in \{0,1\}^n$, we will choose $h_1, \ldots, h_k$ so that $(y, h_1, \ldots, h_k)$ will have about the same distribution as it was drawn from D_{Uni}. To do so, we choose $h_1, \ldots, h_k$ one by one, each time we keep sampling until we find an hash function that its value on y is consistent with A's answer (if the answer is inconsistent, we "rewind" A to its state before it was asked the last "faulty" hash function). We call D_{Sim} the distribution the above process induces on $(y, h_1, \ldots, h_k)$. Assuming that we could prove that the statistical difference between D_{Uni} and D_{Sim} is smaller than $\frac{\varepsilon^2}{n^c}$ (recall that this is the inverting probability of the naive algorithm on D_{Uni}), we could easily conclude the proof of the binding property. Unfortunately, we cannot prove such a strong bound.

Note that till now we did not take advantage of the full powers of A, since we did not use the fact that A finds two different outputs of f that are consistent with the protocol and not merely a single one (indeed the above observations hold also w.r.t. the honest S). When taking into account the full powers of A, we manage to prove that the success probability of the naive algorithm w.r.t. D_{Uni} does not depend on inverting too few elements. More specifically, the subset of $y \in \mathsf{Consist}(h_1, \ldots, h_k)$

[6] With probability $|\mathsf{Consist}(h_1, \ldots, h_m)| / 2^n$, a random y is uniformly distributed in $\mathsf{Consist}(h_1, \ldots, h_m)$. Now if $f(x_1)$ or $f(x_2)$ are in $\mathsf{Consist}(h_1, \ldots, h_m)$ (which clearly happens with probability at least ε), then with probability at least $1/|\mathsf{Consist}(h_1, \ldots, h_m)|$ either x_1 or x_2 are the inverse of y. All in all, we invert y with probability $\frac{|\mathsf{Consist}(\mathcal{H}_1, \ldots, h_m)|}{2^n} \cdot \varepsilon \cdot \frac{1}{|\mathsf{Consist}(h_1, \ldots, h_m)|} = \frac{\varepsilon}{2^n}$.

such that the naive algorithm inverts on them with "high enough" probability is of relative size $\sqrt{\varepsilon \cdot |\mathsf{Consist}(h_1, \ldots, h_k)|}$. [7]

The latter observation turns to be useful, since we also mange to prove the following. For most choices of $h_1, \ldots, h_k$ (excluding a set of probability much smaller than $\frac{\varepsilon^2}{n^c}$), and for most elements in $\mathsf{Consist}(h_1, \ldots, h_k)$ (excluding a set of size much smaller than $\sqrt{\varepsilon \cdot |\mathsf{Consist}(h_1, \ldots, h_k)|}$), the probability mass that $(y, h_1, \ldots, h_k)$ has under D_{Uni} is within a constant factor from its mass under D_{Sim}. By the above observations, it follows that we can invert y with noticeable probability over D_{Sim}, which directly implies that we can invert f (again, with noticeable probability) on the uniform distribution over $\{0,1\}^n$.

1.9 Outline

In Section 3, we generalize the definition of interactive hashing, present our new construction and prove that it satisfies the new definition, where in Section 4 we generalize this result for non-Boolean hash functions. In Section 5, we argue that the new proof can also be applied to the original NOVY protocol (that uses very specific hash functions). Discussion and further issues appear in Section 6. Finally, in Appendix A we show how to use our new theorem to derive a direct construction of statistically hiding commitment based on known regular one-way functions.

2 Preliminaries

2.1 Notations

Given two strings x and y, we denote their concatenation by $x \circ y$. Given a function $f : \{0,1\}^* \mapsto \{0,1\}^*$ and a set $L \subseteq \{0,1\}^*$, we denote the image of f on L as $f(L) \stackrel{\text{def}}{=} \{f(x) \ : \ x \in L\}$ and denote $f(\{0,1\}^*)$ by $Im(f)$. A function $\mu : \mathbb{N} \to [0,1]$ is called *negligible* if $\mu(n) = n^{-\omega(1)}$. We let $\mathrm{neg}(n)$ denote an arbitrary negligible function (i.e., when we say that $f(n) < \mathrm{neg}(n)$ we mean that *there exists* a negligible function $\mu(n)$ such that for every n, $f(n) < \mu(n)$). Likewise, $\mathrm{poly}(n)$ denotes an arbitrary function $f(n) = n^{O(1)}$, where PPT refers to probabilistic algorithms (i.e., Turing machines) that run in *strict* polynomial time.

Given a random variable X taking values in a finite set $\mathcal{U}$, we write $x \leftarrow X$ to indicate that x is selected according to X. Given a subset S of $\mathcal{U}$, we let $x \leftarrow S$ denote that x is selected according to the uniform distribution on S. We adopt the convention that when the same random variable occurs several times in an expression, all occurrences refer to a single sample. For example, $\Pr[f(X) = X]$ is defined to be the probability that when $x \leftarrow X$, we have $f(x) = x$. We write U_n to denote the random variable distributed uniformly over $\{0,1\}^n$. Let D be a distribution over the set L, the support of D is defined as: $\mathrm{Supp}(D) \stackrel{\text{def}}{=} \{x \in L \ : \ D(x) > 0\}$. We write X^k to denote the random variable consisting of k independent copies of X. The statistical distance of two distributions P and Q over Ω, denoted $SD(P,Q)$, is defined as

$$SD(P,Q) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{x \in \Omega} \left| \Pr_P(x) - \Pr_Q(x) \right| \ .$$

[7] Loosely, let T be the set of y's that A is likely to output their inverse (according to f). A random selection of $h_{k+1}, \ldots, h_m$ separates every two elements in T with probability $1 - 2^{-(m-k)}$. So unless the size of T is large enough, one of the two values A output will be forced to be the inverse of an element outside of T. This will contradict the assumptions that values outside of T are only inverted with small probability.

An *interactive protocol* (A, B) consists of two algorithms that compute the next-message functions of the (honest) parties in the protocol. We write $(A(a), B(b))(x)$ to denote the random process obtained by having A and B interact on common input x, with (private) auxiliary inputs a and b to A and B, respectively (if any), and with independent random coin tosses for A and B. We call (A, B) *polynomially bounded* if there is a polynomial p such that for all x, a, b, the running-time each party is at most $p(|x|)$ with probability one. Moreover, if B^* is any interactive algorithm, then A will immediately halt in $(A(a), B^*(b))(x)$ if its running-time exceeds $p(|x|)$; we have the analogous requirement for B interacting with any A^*. The number of *rounds* in an execution of the protocol is the *total* number of messages exchanged between A and B. We let $\text{view}_A(A(a), B(b))(x)$ denotes A's *view* of the interaction, i.e., its values are transcripts $(\gamma_1, \gamma_2, \ldots, \gamma_t; r)$, where the γ_i's are all the messages exchanged and r is A's coin tosses. Similarly, $\text{view}_B(A(a), B(b))$ denotes B's view.

2.2 Pairwise Independent Hash Functions

DEFINITION 2.1 (pairwise Independent hash functions)
Let $\mathcal{H}$ be a family of functions mapping strings of length n to strings of length $\ell(n)$. We say that $\mathcal{H}$ is an efficient family of pairwise independent hash functions (following [CW79]) if the following hold: [8]

Samplable. $\mathcal{H}$ is polynomially samplable in n,

Efficient. There exists a polynomial-time algorithm that given $x \in \{0,1\}^n$ and a description of $h \in \mathcal{H}$ outputs $h(x)$,

Pairwise independence. For every distinct $x_1, x_2 \in \{0,1\}^n$ and every $y_1, y_2 \in \{0,1\}^{\ell(n)}$, we have

$$\Pr_{h \leftarrow \mathcal{H}}[h(x_1) = y_1 \wedge h(x_2) = y_2] = 2^{-2\ell(n)} \ .$$

It is well known ([CW79]) that there exists an efficient family of pairwise-independent hash functions for every $\ell(n) \in \text{poly}(n)$, whose elements description size is $O(\max\{n, \ell(n)\})$.

Given a family of hash functions, we sometime consider the concatenation of this family to itself defined next.

DEFINITION 2.2 (product hash family)
Let $\mathcal{H}$ be a family of functions mapping strings of length n to strings of length $\ell(n)$ and let $k : \mathbb{N} \mapsto \mathbb{N}$. The k-product-family of $\mathcal{H}$, denoted $\mathcal{H}^{k(n)}$, is a family of functions mapping strings of length n to strings of length $k(n)\ell(n)$ which is defined as follows: The members of $\mathcal{H}^{k(n)}$ are all possible tuples $\overline{h}$ of $k(n)$ functions from $\mathcal{H}$. For every such tuple $\overline{h} = (\overline{h}_1, \ldots, \overline{h}_{k(n)})$ and every $x \in \{0,1\}^n$ we let $\overline{h}(x) = (\overline{h}_1(x), \ldots, \overline{h}_{k(n)}(x))$.

The following standard lemma (see for example, [Gol01, Lemma 4.3.1]) states that a random pairwise independent hash function partitions a given set into (almost) equal size subsets.

[8] The first two properties, regarding the efficiency of the family, implicitly assume an ensemble of families (one family for every value of n). For simplify of presentation, we only refer to a single family.

LEMMA 2.3
Let $\mathcal{H}$ be a family of pairwise independent hash functions mapping strings of length n to strings of length ℓ, let $L \subseteq \{0,1\}^n$ and let $\mu = |L|/2^\ell$. Then for every $y \in \{0,1\}^\ell$ and $\delta > 0$, it holds that

$$\Pr_{h \leftarrow \mathcal{H}}[||\{x \in L \ : \ h(x) = y\}| - \mu| > \delta\mu] < \tfrac{1}{\delta^2 \mu} \ .$$

We also make use of the following Corollary.

COROLLARY 2.4
Let $\mathcal{H}$, L and μ be as in Lemma 2.3. Then for every $\delta > 0$ it holds that

$$\Pr_{x \leftarrow L, h \leftarrow \mathcal{H}}[|\{x' \in L \ : \ h(x') = h(x)\}| < \mu(1-\delta)] < \tfrac{1}{\delta^2 \mu} \ .$$

Proof. For $h \in \mathcal{H}$, let $B_h \stackrel{\text{def}}{=} \{y \in \{0,1\}^\ell \ : \ |\{x \in L \ : \ h(x) = y\}| < \mu(1-\delta)\}$. Clearly, for every $h \in \mathcal{H}$ it holds that

$$\Pr_{x \leftarrow L}[h(x) \in B_h] < \frac{|B_h| \cdot \mu}{|L|} = \frac{|B_h|}{2^\ell} = \Pr_{y \leftarrow \{0,1\}^\ell}[y \in B_h] \ .$$

Thus, $\Pr_{x \leftarrow L, h \leftarrow \mathcal{H}}[h(x) \in B_h] < \Pr_{y \leftarrow \{0,1\}^\ell, h \leftarrow \mathcal{H}}[y \in B_h]$. Lemma 2.3 yields that $\Pr_{y \leftarrow \{0,1\}^\ell, h \leftarrow \mathcal{H}}[|\{x \in L \ : \ h(x) = y\}| < \mu(1-\delta)] = \Pr_{y \leftarrow \{0,1\}^\ell, h \leftarrow \mathcal{H}}[y \in B_h] < \frac{1}{\delta^2 \mu}$. We conclude that, $\Pr_{x \leftarrow L, h \leftarrow \mathcal{H}}[|\{x \in L \ : \ h(x) = h(x)\}| < \mu(1-\delta)] = \Pr_{x \leftarrow L, h \leftarrow \mathcal{H}}[h(x) \in B_h] < \frac{1}{\delta^2 \mu}$. □

3 The New Theorem

In this section we present our extended definition for an interactive hashing protocol and give a revised construction and new proof that match this definition.

3.1 Defining a New Notion of Interactive Hashing

We choose (following [NOV06]) to state our definitions in the setting of binary relations. This generalizes the original definition due to [NOVY98], which concentrates on the particular relations that are naturally defined by one-way permutations (see Corollary 3.14). In particular, the underlying relation is not necessarily efficiently computable or even not efficiently verifiable. Moreover, the relation is not necessarily defined over all strings of a given length, but might rather be defined over some small subset of the strings.

DEFINITION 3.1 (interactive hashing)
Let $\mathcal{H}$ be a family of hash functions mapping strings of length n to strings of length $\ell(n)$. An $\mathcal{H}$-interactive hashing $\mathsf{IH} = (\mathsf{S}, \mathsf{R})$ is a probabilistic polynomial-time interactive protocol. Both parties receive the security parameter 1^n as an input and S gets as a private input $y \in \{0,1\}^n$. At the end, S locally outputs y and R outputs $(h, z) \in \mathcal{H} \times \{0,1\}^{\ell(n)}$. We make the following correctness requirement: For all n, all $y \in \{0,1\}^n$, and every pair $(y, (h, z))$ that may be output by $(\mathsf{S}(1^n, y), \mathsf{R}(1^n))$, it is the case that $h(y) = z$.

The security of interactive hashing protocol has two aspects. Binding the sender to y and concealing some information regarding y from R. In this work we focus on security w.r.t. polynomially-bounded sender and unbounded receiver. The setting where both the receiver and the sender are unbounded, called *information theoretic interactive hashing* (a.k.a. *interactive hashing for static sets*), is not treated by this work (for details on the information theoretic setting see for example [CCM98, CS06, DHRS04]). We start by formalizing the binding property.

For a binary relation, W and $y \in \{0,1\}^*$, we denote the set $\{x \in \{0,1\}^* \ : \ W(x,y) = 1\}$ by W_y.

DEFINITION 3.2 (binding)
Let $L \subseteq \{0,1\}^n$, let W be a binary relation, let $\mathcal{H}$ be a family of hash functions mapping strings of length n to strings of length $\ell(n)$ and finally let $\mathsf{IH} = (\mathsf{S}, \mathsf{R})$ be an $\mathcal{H}$-interactive-hashing protocol. We say that IH is (computationally) binding for L and W, if no PPT S^* succeed in the following game with more than negligible probability.

> On security parameter 1^n, S^* interacts with R and R outputs (h, z). Then S^* outputs pairs $(x_0, y_0), (x_1, y_1) \in W$ such that $y_0 \neq y_1 \in L$ and $h(y_0) = h(y_1) = z$.

REMARK 3.3 (comparing to NOVY)
The binding proof of Naor et al. [NOVY98] holds w.r.t. $L = \{0,1\}^n$, W is the relation naturally defined by a one-way permutation and $\mathcal{H}$ is a specific type family of two-to-one Boolean hash functions. See Section 5 for more details on the NOVY protocol.

The following definition states that the only information that a semi-honest receiver acquires through the protocol about y is its hash value for a uniformly chosen hash function.

DEFINITION 3.4 (hiding against semi-honest receivers)
Let $\mathcal{H}$ be a family of hash functions mapping strings of length n to strings of length $\ell(n)$ and let $\mathsf{IH} = (\mathsf{S}, \mathsf{R})$ be an $\mathcal{H}$-interactive-hashing protocol. We say that IH is (statistically) hiding against semi-honest receivers, if there exists a polynomial-time simulator Sim such that for every $y \in \{0,1\}^n$ the distributions $\mathrm{view}^{\pi}_{\mathsf{R}}(\mathsf{S}(y), \mathsf{R})(1^n)$ and $Sim(1^n, h, h(y))_{h \leftarrow \mathcal{H}}$ are identical.

REMARK 3.5 (cheating receivers)
Some level of hiding can be guaranteed by our protocol even against *malicious* R. Specifically, the protocol hides any information regarding the index of y among all the preimages of $z = h(y)$ w.r.t. h. In the setting of [NOVY98] this information is quite meaningful and is also easy to construct. This is because y is chosen uniformly in $\{0,1\}^n$ and regardless of the way the receiver selects h, there are exactly two possible preimages of z. The two preimages can be found easily and therefore the relative index of y is easy to construct. In the most general setting, however, we encounter two problems. Firstly, a malicious R may be able to force the existence of only a single preimage of z w.r.t. h *that lies in* L. Secondly, it may be difficult to find the preimages of z that lie in L. We note that assuming the existence of one-way functions, in several cryptographic applications of interactive hashing (e.g., statistically hiding bit-commitment, see [HHK+05, Theorem 6.1]) any protocol that is secure against an honest receiver can be complied into a protocol that is secure against a malicious receiver.

3.2 The Interactive Hashing Protocol

Let $m\colon \mathbb{N} \mapsto \mathbb{N}$ and let $\mathcal{H}$ be a family of efficiently computable functions defined over strings of length n. We define our interactive-hashing protocol as follows.

PROTOCOL 3.6

Interactive-hashing protocol $\mathsf{IH} = (\mathsf{S}, \mathsf{R})$.

Common input: 1^n.

S's inputs: $y \in \{0,1\}^n$.

1. For $i = 1$ to $m(n)$:
   (a) R chooses uniformly at random $h_i \in \mathcal{H}$ and sends its description to S.
   (b) S sends $z_i = h_i(y)$ back to R.
2. S locally outputs y.
3. R outputs $(\overline{h}, \overline{z}) = ((h_1, \ldots, h_{m(n)}), (z_1, \ldots, z_{m(n)}))$.

The following lemma is immediate from Definitions 3.1 and 3.4.

LEMMA 3.7

Let $\mathcal{H}^{m(n)}$ be the $m(n)$-product-family of $\mathcal{H}$, then IH is hiding against semi-honest receivers, $\mathcal{H}^{m(n)}$-interactive-hashing protocol.

3.3 The Main Theorem - Binding

THEOREM 3.8

Let W be a binary relation and let $L \subseteq \{0,1\}^n$. Let $\mathcal{H}$ be an efficient family of pairwise independent Boolean hash functions defined over strings of length n and let $m(n) \leq \lfloor \log(|L|) \rfloor$. Finally, let IH be the instantiation of Protocol 3.6 with the above $\mathcal{H}$ and m, and let A be an algorithm that runs in time $t_A(n)$ and breaks the binding of IH w.r.t. W and L with probability $\varepsilon_A(n)$.

Then there exists an oracle algorithm $M^{(\cdot)}$ that given an oracle access to A, the following holds for large enough n.

$$\Pr_{y \leftarrow L}[M^A(y) \in W_y] \in \Omega\left(\frac{2^{m(n)}}{|L|} \cdot \frac{\varepsilon_A(n)^2}{n^8}\right) .$$

Letting $t_{\mathcal{H}}(n)$ be an upper bound of the sampling and computing time of $\mathcal{H}$, the running-time of M^A is $O(\log(n)(t_A(n) + m(n)t_{\mathcal{H}}(n)))$.

REMARK 3.9

Note that the binding for $m(n) > \lfloor \log(|L|) \rfloor$ immediately follows by the binding of the first $\lfloor \log(|L|) \rfloor$ rounds. We also point out that M^A does not need to know L, W or ε_A.

REMARK 3.10 (comparing the parameters to [NOVY98] and [NOV06])
For $L = \{0,1\}^n$ and $m(n) = n-1$, the success probability of M^A is $\Omega(\frac{\varepsilon_A(n)^2}{n^8})$, where the running-time is still $O(\log(n)t_A(n) + m(n)\log(n)t_{\mathcal{H}}(n))$. We point that the same success probability and running-time apply also for the NOVY protocol (see Section 5 for details). This is an improvement in parameters compared with the analysis in [NOV06, LemmaB.2]. There the algorithm runs in time $O(nT_A(n) + mn \cdot t_{\mathcal{H}}(n))$ and breaks f with probability $\Omega(\frac{\varepsilon_A(n)^3}{n^6})$. Finally, in the [NOVY98, [Lemma 2] analysis, the algorithm runs in time $O(nT_A(n) + mn \cdot t_{\mathcal{H}}(n))$ (same as in [NOV06]) and only guarantees to break f with probability $\Omega(\frac{\varepsilon_A(n)^{10}}{n^8})$.

The following corollaries follow Lemma 3.7 and Theorem 3.8.

DEFINITION 3.11
Let W be a relation and let $L \subseteq \{0,1\}^n$. We say that W is hard-to-satisfy over L if for any PPT A it holds that $\Pr_{y \leftarrow L}[A(y) \in W_y] < \text{neg}(n)$.

COROLLARY 3.12
Let L, m, $\mathcal{H}^{m(n)}$, W and IH be as in Theorem 3.8. Assuming that W is hard-to-satisfy over L and that $m(n) \geq \log(|L|) - O(\log(n))$, then IH is binding and hiding, against semi-honest receivers, for L and W.

DEFINITION 3.13
Let $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ be an efficiently computable function and let $L \subseteq \{0,1\}^{\ell(n)}$. We say that f is hard to invert over L if for any PPT A it holds that $\Pr_{y \leftarrow L}[A(y) \in f^{-1}(y)] < \text{neg}(n)$.

COROLLARY 3.14
Let L, m, $\mathcal{H}^{m(n)}$ and (S,R) be as in Theorem 3.8 and let $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$. Assuming that f is hard to invert over L and that $m(n) \leq \log(|L|) - O(\log(n))$, then IH is binding and hiding, against semi-honest receivers, for L and $W \stackrel{\text{def}}{=} \{(x, f(x)) \ : \ x \in \{0,1\}^n\}$.

Proof. (of Theorem 3.8) We assume without lost of generality that A is deterministic. Given a randomized algorithm A that breaks the binding of IH when instantiated with a family of hash functions $\mathcal{H}$, consider the family of hash function $\mathcal{H}'$ obtained from $\mathcal{H}$ by appending random strings of length $t_A(n)$ to the hash functions' description. Clearly, the deterministic algorithm A' that emulates A using the randomness given in the hash functions description, breaks the binding of IH instantiated with $\mathcal{H}'$, with exactly the same probability as A does w.r.t. $\mathcal{H}$. In the following proof we assume nothing about $\mathcal{H}$ but being pairwise independent. Thus, the assumption that A is deterministic is indeed without lost of generality.

For simplicity we drop the dependency on n whenever it is clear from the context. We use throughout the proof the following random variables: For $k \in [m]$ and $\overline{h} \in \mathcal{H}^k$, let $A^{Com}(\overline{h}) \in \{0,1\}^k$ be A's answers when questioned by $\overline{h}$ and let $\mathsf{Consist}(\overline{h}) = \left\{y \in L \ : \ \overline{h}(y) = A^{Com}(\overline{h})\right\}$ (i.e., the set of y's that are consistent with A's answers). Finally, we assume without loss of generality that following any sequence of questions $\overline{h} \in \mathcal{H}^m$, A outputs two pairs of elements $(x_0,y_0),(x_1,y_1) \in \{0,1\}^* \times \{0,1\}^n$ and denote them by $A^{Dec}(\overline{h})$. For $\text{ofs} = \max\{m, \lceil 8\log(n) + \log(1/\varepsilon_A)\rceil + 13\}$, we

consider the following algorithm for satisfying W on L.

ALGORITHM 3.15

Algorithm M^A.

Input: $y \in L$

1. Let $\overline{h} \leftarrow \mathsf{Searcher}(y)$.
2. Return $\mathsf{Inverter}(\overline{h})$.

Algorithms Searcher and Inverter are defined as follows.

ALGORITHM 3.16

Algorithm Searcher.

Inputs: $y \in L$.

1. For $k = 1$ to $m - \text{ofs}$:
   Do the following $2\log(n)$ times:
   (a) Set a value for h_k uniformly at random in $\mathcal{H}$.
   (b) If $A^{Com}(h_1, \ldots, h_k)_k = h_k(y)$, break the inner loop.
2. Return $(h_1, \ldots, h_{m-\text{ofs}})$.

ALGORITHM 3.17

Algorithm Inverter.

Inputs: $\overline{h} \in \mathcal{H}^{m-\text{ofs}}$.

1. Choose uniformly at random $\overline{h^e} \in \mathcal{H}^{\text{ofs}}$.
2. Set $((x_0, y_0), (x_1, y_1)) \leftarrow A^{Dec}(\overline{h}, \overline{h^e})$.
3. Return x_0 with probability half and x_1 otherwise.

REMARK 3.18

The value ofs depends in our proof on ε_A. This seems to contradict Remark 3.9 that M^A does not need to know ε_A. Nevertheless, ofs can instead be selected at random with only a factor m decrease in the success probability of M^A. More interestingly, setting ofs $= 0$ will also guarantee M^A the succuss probability claimed in the theorem. The only affect of decreasing ofs to zero is that $\overline{h^e}$ will be selected by the rewinding method of Searcher rather than uniformly at random by Inverter. For every value $\overline{h^e}$ that satisfies $y \in \mathsf{Consist}(\overline{h}, \overline{h^e})$, we have that the probability of selecting it with the rewinding technique is only larger than the probability of uniformly selecting it. A value of $\overline{h^e}$ such that $y \notin \mathsf{Consist}(\overline{h}, \overline{h^e})$ will not contribute in our analysis to the success probability of M^A. It follows that the distinction between Searcher and Inverter is not necessary for the proof. Still, following [NOVY98], we find this distinction very useful for pedagogical reasons.

Assuming that we use the proper data structure to support the rewinding action, it follows that the running time of M^A is $O(\log(n)t_A(n) + m\log(n)\cdot t_{\mathcal{H}}(n))$. We assume without loss of generality that $m \geq \lceil 8\log(n) + \log(1/\varepsilon_A)\rceil + 13$, otherwise we can set $\mathsf{ofs} = m$ and conclude the proof of the theorem directly as follows.

$$\begin{aligned}
&\Pr_{y\leftarrow L}[M^A(y) \in W_y] = \sum_{y\in L}\frac{1}{|L|}\cdot \Pr[\mathsf{Inverter}(H^m) \in W_y] \\
\geq\ & \frac{1}{|L|}\sum_{y\in L}\frac{1}{2}\cdot \Pr\left[((x_0,y_0),(x_1,y_1)) \leftarrow A^{Dec}(H^m)\ :\ x_0 \in W_y \vee x_1 \in W_y\right] \\
\geq\ & \frac{\varepsilon_A}{|L|} \in \Omega\left(\frac{2^m}{|L|}\cdot\frac{\varepsilon_A^2}{n^8}\right)
\end{aligned}$$

We consider the success probability of A w.r.t. the following two distributions.

- $D_{\mathrm{Sim}} \stackrel{\mathrm{def}}{=} \left(\overline{h}, y\right)_{y\leftarrow L, \overline{h}\leftarrow \mathsf{Searcher}(y)}$
- $D_{\mathrm{Uni}} \stackrel{\mathrm{def}}{=} \left(\overline{h}, y\right)_{\overline{h}\leftarrow \mathcal{H}^{m-\mathsf{ofs}}, y\leftarrow \mathsf{Consist}(\overline{h})}$

Given that y is uniformly chosen in L, then D_{Sim} is the distribution that $\mathsf{Inverter}$ is invoked upon through the execution of M^A. Thus, the probability that $\mathsf{Inverter}$ satisfies W over D_{Sim} equals to the success probability of M^A. On the other hand, it is rather easy to show that the probability that $\mathsf{Inverter}$ satisfies W over D_{Uni} is noticeable (as a function of ε_A). Intuitively, this is because the distribution of $\overline{h}$ in D_{Uni} is uniform and this is also the distribution of $\overline{h}$ that A encounters when interacting with R. Proving that, however, does not suffice to deduce that the success probability of $\mathsf{Inverter}$ over D_{Sim} is also high. The reason is that potentially the success probability of $\mathsf{Inverter}$ over D_{Uni} could stem from a relatively few elements that have significantly smaller probability mass w.r.t. D_{Sim} than w.r.t. D_{Uni}. To overcome this problem, we prove (Lemma 3.19) that the probability that $\mathsf{Inverter}$ satisfies W over D_{Uni} is well spread. Even if we ignore the contribution to the success probability of some sufficiently small number of values in the support of D_{Uni}, this success probability will remain noticeable. Having that, we are guaranteed that the success probability of $\mathsf{Inverter}$ is high w.r.t. any distribution that assigns about the same mass to *most* elements in $\mathrm{Supp}(D_{\mathrm{Uni}})$. We then show (Lemma 3.20) that D_{Sim} satisfies this property.

Let us turn to a more formal discussion. For $\overline{h} \in \mathcal{H}^{m-\mathsf{ofs}}$ we let $\varepsilon_{\overline{h}} = \Pr[A \text{ breaks the binding of } \mathsf{IH} \mid (h_1,\ldots,h_{m-\mathsf{ofs}}) = \overline{h}]$, where $h_1,\ldots,h_{m-\mathsf{ofs}}$ are the hash functions chosen by R in the execution of IH. We define the ***weight*** of y w.r.t. $\overline{h}$, by $w(y \mid \overline{h}) = \frac{1}{2}\Pr[A \text{ breaks the binding of } \mathsf{IH} \text{ outputting } ((x_0,y_0),(x_1,y_1)) \wedge y \in \{y_0,y_1\} \mid (h_1,\ldots,h_{m-\mathsf{ofs}}) = \overline{h}]$. Note that $w(y \mid \overline{h})$ is a lower bound on the probability that $\mathsf{Inverter}$ satisfies W on y conditioned on $(h_1,\ldots,h_{m-\mathsf{ofs}}) = \overline{h}$. Finally, for a given set $V \subseteq \mathcal{H}^{m-\mathsf{ofs}} \times \{0,1\}^n$, we let $w_{\overline{V}}(y \mid \overline{h})$ be zero if $(\overline{h}, y) \in V$ and $w(y \mid \overline{h})$ otherwise.

In the following we prove two lemmata w.r.t. the above measures. The first lemma states that the success probability of A over D_{Uni} does not come from small sets. Where the second lemma complete the picture by stating that D_{Sim} approximates D_{Uni} well over all elements in $\mathrm{Supp}(D_{\mathrm{Uni}})$, save but, maybe, a small set.

LEMMA 3.19
Let $V \subseteq \mathrm{Supp}(D_{\mathrm{Uni}})$ such that $\Pr\left[\left|\mathsf{Consist}(H^{m-\mathrm{ofs}}) \cap V\right| > \sqrt{2^{\mathrm{ofs}-1}\varepsilon_{H^{m-\mathrm{ofs}}}}\right] < \varepsilon_A/2$, then $\mathsf{Ex}_{(\overline{h},y)\leftarrow D_{\mathrm{Uni}}}\left[w_{\overline{V}}(y \mid \overline{h})\right] \in \Omega(\varepsilon_A 2^{m-\mathrm{ofs}}/|L|)$.

LEMMA 3.20
There exists a set $V \subseteq \mathrm{Supp}(D_{\mathrm{Uni}})$ such that the following hold:

1. For every $(\overline{h},y) \in \mathrm{Supp}(D_{\mathrm{Uni}})\backslash V$ it holds that $\frac{1}{81} \leq \frac{D_{\mathrm{Sim}}(\overline{h},y)}{D_{\mathrm{Uni}}(\overline{h},y)} \leq 81$,
2. $\Pr\left[\left|\mathsf{Consist}(H^{m-\mathrm{ofs}}) \cap V\right| > 54n^4\right] \in \Omega(n^3 2^{m-\mathrm{ofs}}/|L|)$.

Before proving the above lemmata, let us first use the them for proving Theorem 3.8. By a Markov argument it follows that the success probability of A is at least $\frac{\varepsilon_A}{2}$, even if it is "forced" to fail on every $\overline{h} \in \mathcal{H}^{m-\mathrm{ofs}}$ such that $\varepsilon_{\overline{h}} < \frac{\varepsilon_A}{2}$. Therefore, we assume without loss of generality that either $\varepsilon_{\overline{h}} = 0$ or $\varepsilon_{\overline{h}} \geq \frac{\varepsilon_A}{2}$ and thus $\sqrt{2^{\mathrm{ofs}-1}\varepsilon_{\overline{h}}} > \sqrt{2^{\mathrm{ofs}-2}\varepsilon_A} > 54n^4$, for every $\varepsilon_{\overline{h}} > 0$. Let V be the set whose existence is guaranteed by Lemma 3.20. The definition of $w(\cdot)$ implies that $\Pr[\mathsf{Inverter}(\overline{h}) \in W_y] \geq w(y \mid \overline{h})$. Thus, $\Pr_{y\leftarrow L}[M^A(y) \in W_y] = \Pr_{(\overline{h},y)\leftarrow D_{\mathrm{Sim}}}[\mathsf{Inverter}(\overline{h}) \in W_y] \geq \mathsf{Ex}_{(\overline{h},y)\leftarrow D_{\mathrm{Sim}}}[w_{\overline{V}}(y \mid \overline{h})]$. Where since D_{Uni} approximates D_{Sim} well on any element outside V, it follows that $\Pr_{y\leftarrow L}[M^A(y) \in W_y] \geq \frac{1}{81}\mathsf{Ex}_{(\overline{h},y)\leftarrow D_{\mathrm{Uni}}}[w_{\overline{V}}(y \mid \overline{h})]$. Finally, Lemma 3.19 yields that, $\Pr_{y\leftarrow L}[M^A(y) \in W_y] \in \Omega(\frac{2^m}{|L|} \cdot \frac{\varepsilon_A^2}{n^8})$. □

3.3.1 Proving Lemma 3.19

We start by noticing that in each step of the protocol, the number of elements inside L that are consistent with the transcript so far is w.h.p. (regardless of A's answers) not faraway from the expected value.

DEFINITION 3.21
Let $k \in [m]$. We call $\overline{h} \in \mathcal{H}^k$ balanced, if for every $j \in [k]$ it holds that $\frac{|L|}{3\cdot 2^j} \leq \left|\mathsf{Consist}(\overline{h}_{1,\ldots,j})\right| \leq \frac{3\cdot|L|}{2^j}$.

CLAIM 3.22
For every $k \in [m-\mathrm{ofs}]$ it holds that $\Pr[H^k \text{ is balanced}] \geq 1 - 6n^2 2^k/|L|$.

Proof. We say that $h \in \mathcal{H}$ ***well partitions*** the set $\mathsf{Consist}(\overline{h})$, if $\left|\mathsf{Consist}(\overline{h},h)\right| \in \left[(\frac{1}{2}-\frac{1}{2n})\cdot\left|\mathsf{Consist}(\overline{h})\right|, (\frac{1}{2}+\frac{1}{2n})\cdot\left|\mathsf{Consist}(\overline{h})\right|\right]$. Let $\overline{h} \in \mathcal{H}^k$, it is easy to verify that if $\overline{h}_j$ well partitions $\mathsf{Consist}(\overline{h}_{1,\ldots,j-1})$ for every $j \in [k]$, then $\overline{h}$ is balanced. By Lemma 2.3, for every $j \in [k]$ it holds that $\mathrm{P}[H \text{ does not well partition } \overline{h}_{1,\ldots,j-1}] < 2\left|\mathsf{Consist}(\overline{h}_{1,\ldots,j-1})\right|/n^2$. Therefore, we can lower bound the probability that H^k is

balanced as follows.

$$\begin{aligned}
&\Pr[H^k \text{ is not balanced}] \\
\leq\ &\sum_{j=1}^{k} \Pr[H_j^k \text{ does not well partition } \mathsf{Consist}(H^k_{1,\ldots,j-1}) \mid H^k_{1,\ldots,j-1} \text{ is balanced}] \\
\leq\ &\sum_{j=0}^{k-1} 3n^2 2^j / |L| \leq 6n^2 2^k / |L| \ .
\end{aligned}$$

□

Having the above we are ready to prove Lemma 3.19.

Proof. (of Lemma 3.19) Let's fix for a moment $\overline{h} \in \mathcal{H}^{m-\text{ofs}}$. We assume for simplicity a non-increasing order on the elements of $\mathsf{Consist}(\overline{h})$ according to their weights (i.e., by $w(\cdot \mid \overline{h})$), and denote by $\mathsf{Consist}(\overline{h})_i$ the i^{th} element of $\mathsf{Consist}(\overline{h})$ by this order. The following claim states that the weight is not concentrated only on the first $\ell_{\overline{h}} \stackrel{\text{def}}{=} \left\lfloor \sqrt{2^{\text{ofs}-1}\varepsilon_{\overline{h}}} \right\rfloor$ heaviest elements of $\mathsf{Consist}(\overline{h})$.

CLAIM 3.23
It holds that $\sum_{i=\ell_{\overline{h}}+1}^{|\mathsf{Consist}(\overline{h})|} w(\mathsf{Consist}(\overline{h})_i \mid \overline{h}) \geq \varepsilon_{\overline{h}}/4$.

Proof. Let $Z = \left\{\mathsf{Consist}(\overline{h})_1, \ldots, \mathsf{Consist}(\overline{h})_{\ell_{\overline{h}}}\right\}$, by the pairwise independence of $\mathcal{H}$ it follows that,

$$\Pr[\exists y_0 \neq y_1 \in Z \ :\ H^{\text{ofs}}(y_0) = H^{\text{ofs}}(y_1)] \leq \frac{|Z|^2}{2^{\text{ofs}}} \leq \frac{2^{\text{ofs}}\varepsilon_{\overline{h}}}{2 \cdot 2^{\text{ofs}}} = \varepsilon_{\overline{h}}/2 \qquad (1)$$

Let y_0 and y_1 be the pair of elements returned by A on a successful cheat. Equation 1 yields that the probability that both y_0 and y_1 are inside Z is at most $\varepsilon_{\overline{h}}/2$. It follows that the probability that A cheats successfully while at least one of y_0 and y_1 is outside Z is at least $\varepsilon_{\overline{h}}/2$. Note that each event where A cheats successfully and outputs an element $y_i = y$, contributes half its probability to the total weight of y. Thus, the sum of weights of the elements in $\mathsf{Consist}(\overline{h}) \backslash Z$ is at least $\varepsilon_{\overline{h}}/4$. □

Assuming that $\left|\mathsf{Consist}(\overline{h}) \cap V\right| \leq \sqrt{2^{\text{ofs}-1}\varepsilon_{\overline{h}}}$, Claim 3.23 yields that $\sum_{y \in \mathsf{Consist}(\overline{h})} w_{\overline{V}}(y \mid \overline{h}) \geq \frac{\varepsilon_{\overline{h}}}{4}$. In the following we concentrate on the set Typical $\subseteq \mathcal{H}^{m-\text{ofs}}$ defined as Typical $= \left\{\overline{h} \in \mathcal{H}^{m-\text{ofs}} \ :\ \left|\mathsf{Consist}(\overline{h}) \cap V\right| \leq \sqrt{2^{\text{ofs}-1}\varepsilon_{\overline{h}}} \bigwedge \left|\mathsf{Consist}(\overline{h})\right| \leq 3\,|L|\,/2^{m-\text{ofs}}\right\}$. Claim 3.22 and the assumption about V yield that

$$\Pr[H^{m-\text{ofs}} \notin \text{Typical})] \leq \varepsilon_A/2 + O(n^2 2^{m-\text{ofs}}/|L|) \leq \frac{3 \cdot \varepsilon_A}{4} \qquad (2)$$

It follows that

$$\underset{(\overline{h},y)\leftarrow D_{\mathrm{Uni}}}{\mathsf{Ex}}\left[w_{\overline{V}}(y \mid \overline{h})\right] = \frac{1}{|\mathcal{H}^{\mathrm{ofs}}|}\sum_{\overline{h}\in\mathcal{H}^{m-\mathrm{ofs}}}\frac{1}{|\mathsf{Consist}(\overline{h})|}\sum_{y\in\mathsf{Consist}(\overline{h})} w_{\overline{V}}(y \mid \overline{h})$$

$$\geq \frac{1}{|\mathcal{H}^{\mathrm{ofs}}|}\cdot\frac{2^{m-\mathrm{ofs}}}{3\,|L|}\sum_{\overline{h}\in\mathrm{Typical}}\sum_{y\in\mathsf{Consist}(\overline{h})} w_{\overline{V}}(y \mid \overline{h}) \ .$$

While Claim 3.23 together with the second property of pairs in Typical yield that

$$\underset{(\overline{h},y)\leftarrow D_{\mathrm{Uni}}}{\mathsf{Ex}}\left[w_{\overline{V}}(y \mid \overline{h})\right] \geq \frac{2^{m-\mathrm{ofs}}}{12\,|L|}\cdot\frac{1}{|\mathcal{H}^{\mathrm{ofs}}|}\sum_{\overline{h}\in\mathrm{Typical}}\varepsilon_{\overline{h}} \ .$$

We conclude, using (2), that

$$\underset{(\overline{h},y)\leftarrow D_{\mathrm{Uni}}}{\mathsf{Ex}}\left[w_{\overline{V}}(y \mid \overline{h})\right] \geq \frac{2^{m-\mathrm{ofs}}}{12\,|L|}\cdot\frac{\varepsilon_A}{4} \in \Omega(\varepsilon_A 2^{m-\mathrm{ofs}}/\,|L|) \ .$$

□

3.3.2 Proving Lemma 3.20

We bridge between D_{Uni} and D_{Sim} using the following hybrid distributions. Let $k \in \{0,\ldots,m-1\}$ and let $\overline{h} \in \mathcal{H}^k$. We define the hybrid algorithm $\mathsf{Searcher}^{\overline{h}}(y)$ that sets its first k hash functions to $\overline{h}$ and then continues as the original $\mathsf{Searcher}$ algorithm does, and use it to define the following distributions.

- $D^{\overline{h}}_{\mathrm{Sim}} \stackrel{\mathrm{def}}{=} (y,h)_{y\leftarrow\mathsf{Consist}(\overline{h}),h\leftarrow\mathsf{Searcher}^{\overline{h}}(y)_{k+1}}$
- $D^{\overline{h}}_{\mathrm{Uni}} \stackrel{\mathrm{def}}{=} (y,h)_{h\leftarrow\mathcal{H},y\leftarrow\mathsf{Consist}(\overline{h},h)}$

The proof of Lemma 3.20 easily follows by the next lemma that relates $D^{\overline{h}}_{\mathrm{Sim}}$ to $D^{\overline{h}}_{\mathrm{Uni}}$.

LEMMA 3.24
Let $k \in \{0,\ldots,m-\mathrm{ofs}-1\}$ and let $\overline{h} \in \mathcal{H}^k$ be balanced. Then there exists a set $\mathrm{Bad}(\overline{h}) \subseteq \mathsf{Consist}(\overline{h})$ of size at most $54n^3$ such that the following holds:

$$\Pr\left[\exists y \in \mathsf{Consist}(\overline{h},H)\setminus\mathrm{Bad}(\overline{h}) \ : \ D^{\overline{h}}_{\mathrm{Sim}}(y,H)/D^{\overline{h}}_{\mathrm{Uni}}(y,H) \notin [1-\frac{4}{n},1+\frac{4}{n}]\right] \in O(n^2 2^k/\,|L|) \ .$$

Before proving Lemma 3.24, let us first use it for proving Lemma 3.20.

Proof. (of Lemma 3.20) For $\overline{h} \in \mathcal{H}^k$ let $\mathrm{Diff}(\overline{h}) = \{y \in \mathsf{Consist}(\overline{h}) : \exists i \in [m-\mathrm{ofs}] : D^{\overline{h}_{1,\ldots,i-1}}_{\mathrm{Sim}}(y,\overline{h}_i)/D^{\overline{h}_{1,\ldots,i-1}}_{\mathrm{Uni}}(y,\overline{h}_i) \notin [1-\frac{4}{n},1+\frac{4}{n}]\}$. By induction, for every $y \in \mathsf{Consist}(\overline{h})\setminus\mathrm{Diff}(\overline{h})$ it holds that $\frac{1}{81} \leq D_{\mathrm{Sim}}(\overline{h},y)/D_{\mathrm{Uni}}(\overline{h},y) \leq 81$. In the following we prove that w.h.p. $\mathrm{Diff}(\overline{h})$ is small. Thus, Lemma 3.20 follows by letting $V = \{(y,\overline{h}) \in \mathrm{Supp}(D_{\mathrm{Uni}}) \ : \ y \in \mathrm{Diff}(\overline{h})\}$.

Lemma 3.24 yields that $\Pr[\left|\mathrm{Diff}(H^{m-\mathrm{ofs}})\right| > 54n^4 \mid H^{m-\mathrm{ofs}} \text{ is balanced}] \leq n \cdot \Omega(n^2 2^k / |L|)$. Where by Claim 3.22 we have that $\Pr[H^{m-\mathrm{ofs}} \text{ is balanced}] \geq 1 - \frac{6n^2 2^{m-\mathrm{ofs}}}{|L|}$. It follows that $\Pr[\left|\mathrm{Diff}(H^{m-\mathrm{ofs}})\right| > 54n^4] \leq n \cdot \Omega(n^2 2^k / |L|) + \frac{6n^2 2^{m-\mathrm{ofs}}}{|L|} \in \Omega(n^3 2^{m-\mathrm{ofs}} / |L|)$. □

Proof. (of Lemma 3.24) Consider the Boolean matrix $T^{\left|\mathsf{Consist}(\overline{h})\right| \times |\mathcal{H}|}$, where $T(y,h) = 1$ if $A^{Com}(\overline{h},h)_{k+1} = h(y)$ and zero otherwise. We identify the indices into T with the set $\mathsf{Consist}(\overline{h}) \times \mathcal{H}$. The distribution $D^{\overline{h}}_{\mathrm{Uni}}$ can be described in relation to T as follows. Choose a random column of T and draw the index of a random one entry from this column (where a "one entry" is simply an entry of the matrix that is assigned the value one). The distribution $D^{\overline{h}}_{\mathrm{Sim}}$ can also be described in relation to T as follows. Choose a random row of T and for $2\log(n)$ times draw a random entry from this raw. If a one entry is drawn, then choose its index and stop drawing, otherwise select the index of the last drawn entry.

Let us start with an informal discussion. Compare the matrix T with the matrix $\widehat{T}^{\left|\mathsf{Consist}(\overline{h})\right| \times |\mathcal{H}|}$, where $\widehat{T}(y,h) = h(y)$. Note that T can be derived from $\widehat{T}$ by flipping all values in some of its columns (where the column which corresponds to h is flipped whenever $A^{Com}(\overline{h},h)_{k+1} = 0$). By the pairwise independence of $\mathcal{H}$, it follows that most *columns* of $\widehat{T}$ are balanced (have about the same number of zeros and ones) and thus the same holds for T. Hence, the mass that $D^{\overline{h}}_{\mathrm{Uni}}$ assigns to most of the one entries of T is close to $\frac{1}{|\mathcal{H}|} \cdot \frac{2}{\left|\mathsf{Consist}(\overline{h})\right|}$. Using again the pairwise independent of $\mathcal{H}$, we can prove that most *rows* of T are balanced. Hence, the mass that $D^{\overline{h}}_{\mathrm{Sim}}$ assigns to most one entries in T is also close to $\frac{1}{|\mathcal{H}|} \cdot \frac{2}{\left|\mathsf{Consist}(\overline{h})\right|}$. Since the support of $D^{\overline{h}}_{\mathrm{Uni}}$ and the indices set of one entries in T are the same, we conclude that the one indices in a random raw of T (a random $h \in \mathcal{H}$) get about the same mass in $D^{\overline{h}}_{\mathrm{Sim}}$ and in $D^{\overline{h}}_{\mathrm{Uni}}$, and the proof of the Lemma 3.24 follows.

Let us turn to the formal proof. We define the set $\mathrm{Bad}(\overline{h})$ as $\left\{y \in \mathsf{Consist}(\overline{h}) : \Pr[T(H,y) = 1] \notin [\frac{1}{2} - \frac{1}{2n}, \frac{1}{2} + \frac{1}{2n}]\right\}$ and start by showing that $\mathrm{Bad}(\overline{h})$ is indeed small.

CLAIM 3.25

Assuming that $\overline{h}$ is balanced, then $\left|\mathrm{Bad}(\overline{h})\right| < 54n^3$.

Proof. Let $\mathrm{Bad}_{Law}(\overline{h}) = \left\{y \in \mathsf{Consist}(\overline{h}) : \Pr[T(H,y) = 1] < \frac{1}{2} - \frac{1}{2n}\right\}$. We assume that $\left|\mathrm{Bad}_{Law}(\overline{h})\right| > 27n^3$ and derive a contradiction (the proof that $\left|\mathrm{Bad}(\overline{h}) \setminus \mathrm{Bad}_{Law}(\overline{h})\right| < 27n^3$ is analogous). Consider the matrix $T|_{\mathrm{Bad}_{Law}}$ - the restriction of T to the rows $\mathrm{Bad}_{Law}(\overline{h})$. By definition, the rows of $T|_{\mathrm{Bad}_{Law}}$ have more zeros than ones. Hence, the matrix $T|_{\mathrm{Bad}_{Law}}$ itself has more zeros than ones. On the other hand, by the pairwise independence of $\mathcal{H}$ it follows that most columns of $T|_{\mathrm{Bad}_{Law}}$ are balanced (have about the same number of zeros of ones). Therefore, $T|_{\mathrm{Bad}_{Law}}$ itself is balanced and a contradiction is derived. More formally, for $h \in \mathcal{H}$ let T_h be the number of ones in the h column, that is $T_h = \sum_{y \in \mathrm{Bad}_{Law}(\overline{h})} T(y,h)$. We upper bound the expectation of T_h as follows,

$$\mathsf{Ex}[T_H] = \mathsf{Ex}\Big[\sum_{y \in \mathrm{Bad}_{Law}(\overline{h})} T(y,H)\Big] = \sum_{y \in \mathrm{Bad}_{Law}(\overline{h})} \mathsf{Ex}[T(y,H)] < (\frac{1}{2} - \frac{1}{2n}) \left|\mathrm{Bad}_{Law}(\overline{h})\right| \ .$$

Recall that $T(h,y) = 1$ if $A^{Com}(\overline{h},h)_{k+1} = h(y)$ and zero otherwise. Since the set $\mathrm{Bad}_{Law}(\overline{h})$ is large, Lemma 2.3 yields that a random h splits w.h.p. the elements of $\mathrm{Bad}_{Law}(\overline{h})$ into two almost equals size according to their consistency with A's answer on h. That it, $\Pr\left[T_H < \left|\mathrm{Bad}_{Law}(\overline{h})\right| \cdot (\frac{1}{2} - \frac{1}{3n})\right] < \frac{9n^2}{\left|\mathrm{Bad}_{Law}(\overline{h})\right|} \leq \frac{1}{3n}$. Thus,

$$\begin{aligned}\mathsf{Ex}[T_H] &\geq \frac{1}{|H|} \cdot \sum_{h\in\mathcal{H}\ :\ T_h \geq \left|\mathrm{Bad}_{Law}(\overline{h})\right|\cdot(\frac{1}{2}-\frac{1}{3n})} \left|\mathrm{Bad}_{Law}(\overline{h})\right| \cdot (\frac{1}{2} - \frac{1}{3n}) \\ &> (1-\frac{1}{3n}) \cdot \left|\mathrm{Bad}_{Law}(\overline{h})\right| \cdot (\frac{1}{2} - \frac{1}{3n}) > \left|\mathrm{Bad}_{Law}(\overline{h})\right| \cdot (\frac{1}{2} - \frac{1}{2n}) \ ,\end{aligned}$$

and a contradiction is derived. □

The definition of $\mathrm{Bad}(\overline{h})$ yields that $\Pr_{(y,h)\leftarrow D^{\overline{h}}_{\mathrm{Sim}}}[A^{Com}(\overline{h},h)_{k+1} = h(y) \mid y \notin \mathrm{Bad}(\overline{h})] > 1 - O(1/n^2)$. Thus, for every $h \in \mathcal{H}$ and every $y \in \mathsf{Consist}(\overline{h},h) \setminus \mathrm{Bad}(\overline{h})$, it holds that $D^{\overline{h}}_{\mathrm{Sim}}(y,h) \in [(1-O(1/n^2))\gamma \cdot \frac{1}{1+\frac{1}{n}}, \gamma \cdot \frac{1}{1-\frac{1}{n}}]$, for $\gamma = \frac{2}{|\mathsf{Consist}(\overline{h})|\cdot|\mathcal{H}|}$. Now let $\mathcal{H}^{\mathrm{Bad}}(\overline{h}) = \{h \in \mathcal{H} : \Pr_{y\leftarrow\mathsf{Consist}(\overline{h})}[T(h,y)=1] \notin [\frac{1}{2} - \frac{1}{2n}, \frac{1}{2} + \frac{1}{2n}]\}$. Clearly for every $h \in \mathcal{H} \setminus \mathcal{H}^{\mathrm{Bad}}(\overline{h})$ and every $y \in \mathsf{Consist}(\overline{h},h)$, it holds that $D^{\overline{h}}_{\mathrm{Uni}}(y,h) \in [\gamma\cdot\frac{1}{1+\frac{1}{n}}, \gamma\cdot\frac{1}{1-\frac{1}{n}}]$. It follows that $D^{\overline{h}}_{\mathrm{Sim}}(y,h)/D^{\overline{h}}_{\mathrm{Uni}}(y,h) \in [1-\frac{4}{n}, 1+\frac{4}{n}]$ for every $h \in \mathcal{H} \setminus \mathcal{H}^{\mathrm{Bad}}(\overline{h})$ and $y \in \mathsf{Consist}(\overline{h},h) \setminus \mathrm{Bad}(\overline{h})$, and the proof of Lemma 3.24 follows by next claim.

CLAIM 3.26
Assuming that $\overline{h}$ is balanced, then $\Pr[H \in \mathcal{H}^{\mathrm{Bad}}(\overline{h})] \in \Omega(n^2 2^k/|L|)$.

Proof. Immediate by the pairwise independence of $\mathcal{H}$ (see Lemma 2.3) □

□

4 Protocols with Better Round Complexity

When Protocol 3.6 is invoked with Boolean hash functions (as in Theorem 3.8), it suffers from a linear round complexity. Unfortunately when considering relations that are hard to satisfy by polynomial-time algorithms (see Definition 3.11), Haitner et al. [HHRS07] show that, at least as far as (fully) black-box reductions are concerned, this high number of rounds is unavoidable. [9] Having the above, we consider more efficient protocols whose binding hold w.r.t. relations that are harder to satisfy. In particular, we consider Protocol 3.6 invoked with hash functions whose output length is roughly "log the security" of a given relation, and prove the following theorem.

[9] [HHRS07] show that for a relation W defined by a one-way permutation f (i.e., $W = \{(x,f(x)) : x \in \{0,1\}^n\}$), every binding interactive hashing protocol has $\Omega(n/\log n)$ rounds. While Protocol 3.6 has linear number of rounds, it is straight forward to give a variant of this protocol with only $O(n \log n)$ rounds, in particular such a protocol follows by the main result of this section.

THEOREM 4.1

Let W be a binary relation and let $L \subseteq \{0,1\}^n$. Let $\mathcal{H}$ be an efficient family of pairwise independent hash functions from strings of length n to strings of length $s(n)$ and let $m(n) \leq \left\lfloor \frac{\log(|L|)}{s(n)} \right\rfloor$. Finally, let IH be the instantiation of Protocol 3.6 with the above $\mathcal{H}$ and m, and let A be an algorithm that runs in time $t_A(n)$ and breaks the binding of IH w.r.t. W and L with probability $\varepsilon_A(n)$.

Then there exists an oracle algorithm $M^{(\cdot)}$ that given an oracle access to A, the following holds for large enough n.

$$\Pr_{y \leftarrow L}[M^A(y) \in W_y] \in \Omega\left(\frac{2^{m(n)}}{|L|} \cdot \frac{\varepsilon_A(n)^2}{2^{2s(n)} n^8}\right) .$$

Letting $t_{\mathcal{H}}(n)$ be an upper bound of the sampling and computing time of $\mathcal{H}$, the running-time of M^A is $O(\log(n) 2^{s(n)}(t_A(n) + m(n) t_{\mathcal{H}}(n)))$.

REMARK 4.2

As in Theorem 3.8, the binding for $m(n) > \left\lfloor \frac{\log(|L|)}{s(n)} \right\rfloor$ immediately follows by the binding of the first $\left\lfloor \frac{\log(|L|)}{s(n)} \right\rfloor$ rounds, and M^A does not need to know L, W or ε_A.

Proof's sketch. The proof of Theorem 4.1 follows very closely the proof of Theorem 3.8 and we only point out the main differences. When using hash functions of $s(n)$ bit output rather than Boolean ones, each query made by the receiver partitions the set of consistent elements into $2^{s(n)}$ different subsets (rather than two subsets in the Boolean case). For the second part of the proof to go through (i.e., Lemma 3.20), we need to make sure that the size of each of the subsets induced by a random query is not too far from the expected value. Thus, we need to make sure the initial set of consistent elements is large enough. For the same reason, we cannot guarantee that the set of "bad" elements for which D_{Uni} and D_{Sim} give different weight (i.e., the set V in Lemma 3.20) is very small, but rather have to compensate a much larger set of about $O(n^4 2^{s(n)})$ elements. It turns out that we can adopt Lemma 3.20 for hash functions that output $s(n)$ bits and V of size $O(n^4 2^{s(n)})$, by taking $\text{ofs} \in \Omega\left(\log(n) + \log(1/\varepsilon_A) + 2s(n)\right)$ and change the main loop of the Searcher algorithm (where Searcher tries to find a hash function h such that A's answer on h is equal to $h(y)$) to repeat $2\log(n) \cdot 2^{s(n)}$ times. That is, we get the following lemma.

LEMMA 4.3

There exists a set $V \subseteq \text{Supp}(D_{\text{Uni}})$ such that the following hold:

1. For every $(\overline{h}, y) \in \text{Supp}(D_{\text{Uni}}) \backslash V$ it holds that $\frac{1}{81} \leq \frac{D_{\text{Sim}}(\overline{h},y)}{D_{\text{Uni}}(\overline{h},y)} \leq 81$,
2. $\Pr\left[\left|\text{Consist}(H^{m-\text{ofs}}) \cap V\right| > 54n^4 2^{s(n)}\right] \in \Omega(n^3 2^{m-\text{ofs}} / |L|)$,

where D_{Uni} and D_{Sim} are defined w.r.t. $\text{ofs} = \max\left\{m \cdot s(n), \lceil 8\log(n) + \log(1/\varepsilon_A)\rceil + 13 + 2s(n)\right\}$.

It is left to show that Lemma 3.19 still go through with respect to $s(n)$ bits output hash functions. [10] It is easy to very that the latter holds, however, since in the proof of Lemma 3.19 we

[10] The case of Boolean hash function is immediate, since Lemma 3.19 is stated to accommodate any V whose fraction inside the consistent elements is not too big, which is also the case also w.r.t. the larger set V guaranteed by Lemma 4.3 as the value of ofs we consider is also larger.

did not use the fact the last ofs functions are Boolean and not a single function with ofs bits output. Thus, as in the proof of Theorem 3.8, the proof of Theorem 4.1 easily follows by Lemma 3.19 and Lemma 4.3.

5 Applying Our New Proof to NOVY

In this section we show the proof of Theorem 3.8 can be applied to the following protocol, known as the NOVY protocol, considered by Naor et al. [NOVY98] and Nguyen et al [NOV06].

5.1 The NOVY protocol

For $m(n) \in \mathbb{N}$, we define the following protocol.

PROTOCOL 5.1

The NOVY protocol $\mathsf{IH} = (\mathsf{S}, \mathsf{R})$.

Common input: 1^n.

S's inputs: $y \in \{0,1\}^n$.

1. For $i = 1$ to $m(n)$:
   (a) R chooses uniformly at random $r_i \in \{0,1\}^{n-i}$ and sends $h_i = 0^{i-1} \circ 1 \circ r_i$ over to S.
   (b) S sends $z_i = \langle h_i, y \rangle_2 \mod 2$ back to R.
2. S locally outputs y.
3. R outputs $(\overline{h}, \overline{z}) = ((h_1, \ldots, h_{m(n)}), (z_1, \ldots, z_{m(n)}))$.

That is, the above protocol is the same as Protocol 3.6, but it uses a special type of Boolean function. For $i \in \{1, \ldots, m(n)\}$ let $\mathcal{H}_i$ be the family of functions induced by the selection of h_i described above.

5.2 The new Theorem

Our goal is to prove the following version of Theorem 3.8.

THEOREM 5.2

Let W be a binary relation, let $m(n) \leq n$ and let (S, R) and $\{\mathcal{H}_i\}_{i=1}^{m(n)}$ be as in Protocol 5.1. Finally, let A be an algorithm that runs in time $t_A(n)$ and breaks the binding of (S, R) w.r.t. W with probability $\varepsilon_A(n)$. Then there exists an oracle algorithm $M^{(\cdot)}$ that given an oracle access to A, the following holds for large enough n.

$$\Pr[M^A(U_n) \in W_{U_n}] \in \Omega\left(\frac{1}{2^{n-m(n)}} \cdot \frac{\varepsilon_A(n)^2}{n^8}\right) .$$

Letting $t_{\mathcal{H}}(n)$ be an upper bound of the sampling and computing time of $\{\mathcal{H}_i\}$, then the running-time of M^A is $O(\log(n) \cdot t_A(n) + m \log(n) \cdot t_{\mathcal{H}}(n))$.

Proof. It is easy to verify that if the families of functions $\{\mathcal{H}_i\}_{i=1}^{n-1}$ would have been pairwise independent, then our proof of Theorem 3.8 would hold in this case as well. [11] The latter, however, does not hold and therefore we have to refine our approach. Fortunately, the proof of the theorem does not require that the families of Boolean hash function to be pairwise independent w.r.t. the initial set of inputs L, but rather to be pairwise independent w.r.t. the elements of the initial set that are consistent with the protocol so far. It turns out that given that the initial set is $\{0,1\}^n$, the families of Boolean hash functions used by NOVY are "pairwise independent enough" on the relevant set and thus essentially the same proof as the one we gave for Theorem 3.8 goes through.

Let's us turn to a more formal discussion. For $k \in [m(n)]$, $\overline{h} \in \mathcal{H}_1 \times \cdots \times \mathcal{H}_k$ and $\overline{z} \in \{0,1\}^k$, let $\mathsf{Consist}(\overline{h}, \overline{z})$ be the set of elements inside $\{0,1\}^n$ that are consistent with $\overline{h}$ and $\overline{z}$ (i.e., $\left\{y \in \{0,1\}^n \ : \ \overline{h}(y) = \overline{z}\right\}$). By induction, it follows that for any possible pair $(\overline{h}, \overline{z})$ and element $y_2 \in \{0,1\}^{n-k}$, there exists a *single* element $y_1 \in \{0,1\}^k$ (which depends on y_2 and on $(\overline{h}, \overline{z})$) such that $y_1 \circ y_2 \in \mathsf{Consist}(\overline{h}, \overline{z})$. Hence, for any $y \in \mathsf{Consist}(\overline{h}, \overline{z})$ there exists exactly one other element $y' \in \mathsf{Consist}(\overline{h}, \overline{z})$ for which $y_{k+2\ldots,n} = y'_{k+2\ldots,n}$. Thus, for any other element $y'' \in \mathsf{Consist}(\overline{h}, \overline{z})$, which is different than y and y', it holds that the random variables $h(y)$ and $h(y'')$ (and also $h(y')$ and $h(y'')$), where h is a random function from $\mathcal{H}_{k+1}$, are independent. Therefore, every subset $Z \subseteq \mathsf{Consist}(\overline{h}, \overline{z})$ can be partitioned into two almost equal size subsets (i.e., of difference in size at most one) such that $\mathcal{H}_{k+1}$ is pairwise independent w.r.t. both subsets. Through the proof of Theorem 3.8, we use the pairwise independence property of the hash functions to prove that the following holds. Every fixed large enough subset $Z \subseteq \mathsf{Consist}(\overline{h}, \overline{z})$ is partitioned w.h.p. by a random Boolean hash function into two parts of almost the same size. [12] By our previous observation such a partition also happens, with high enough probability, w.r.t. the family $\mathcal{H}_{k+1}$. Thus, the proof of Theorem 3.8 applied also for the NOVY protocol. □

6 Conclusions

One interesting question is to come with a reduction from interactive hashing to one-way permutations that is even more security preserving. Particularly, is there such a reduction that is linearly-preserving [HL92] (i.e., where the time-success ratio of an adversary inverting the one-way permutation is only larger by a multiplicative polynomial factor than the time-success ratio of an adversary breaking the interactive hashing protocol. There are three possible directions for an improvement: (1) Presenting a more secure protocol than the NOVY protocol (or our variant), (2) Giving a better reduction from an adversary that breaks the interactive hashing to one that breaks the one-way permutations, or (3) Improving the analysis of the reduction mentioned in (2).

Note that our improvement in parameters over the NOVY proof is mainly in the third item (i.e., the analysis of the reduction). In the following we show that our analysis cannot be pushed much further. Namely, we present a (non-efficient) adversary A that breaks the binding of the NOVY protocol with probability ε, but M^A breaks the underlying one-way permutation with probability at most $2 \cdot \varepsilon^{1.4}$.

Consider an algorithm M for inverting a one-way permutation that uses an adversary A of the NOVY protocol in the following black-box manner: On $y \in \{0,1\}^n$, it keeps sampling random hash

[11] Note that the proof of Theorem 3.8 does not require that the same family is used in each round.

[12] Actually, save but the proof of Claim 3.25, we only need this property w.r.t. $Z = \mathsf{Consist}(\overline{h}, \overline{z})$. Note that by the above observation about the structure of $\mathsf{Consist}(\overline{h}, \overline{z})$, every $h \in \mathcal{H}_{k+1}$ *always* partitions $\mathsf{Consist}(\overline{h}, \overline{z})$ into two equal parts.

functions and rewinding A, until it finds a series of $n-1$ hash functions on which A's answers is consistent with y. Then, it returns one of A's outputs as the candidate preimage of y (note that both the NOVY and ours inverting algorithms follow this strategy). Assume that A operates as follows: For $\varepsilon > 0$, it replies with random answers on the first $n - \log(\frac{1}{\varepsilon})$ questions (hash functions) and then randomly selects two distinct elements, $y_1, y_2 \in \{0,1\}^n$, that are consistent with the protocol so far. For the remaining hash functions A does the following: if both y_1 and y_2 yield the same answer then it answers with this value, otherwise, it selects randomly one of the elements and from now on answers according to this element. At the end of the protocol A checks whether both y_1 and y_2 are consistent with the protocol. If the answer is positive, it inverts f on both y_1 and y_2 and outputs the result (recall that the reduction does not assume that A is efficient and therefore it is allowed for example to invert f using exhaustive search), otherwise it outputs $\perp$. Since $\mathcal{H}$ is a family of pairwise independent hash functions, the random variables $h(y_1)$ and $h(y_2)$, for a randomly chosen hash function h, are independent. [13] Thus, the probability that A breaks the NOVY protocol is exactly ε. On the other hand, in order for M to succeed, y has to be selected by A as one of the elements in $\{y_1, y_2\}$. Since the number of elements the are consistent with the protocol after $n - \log(\frac{1}{\varepsilon})$ steps is $1/\varepsilon$, it follows that this happens with probability 2ε. Given that $y \in \{y_1, y_2\}$, say that $y = y_1$, M has to choose in each step an hash function h for which $A(h) = h(y) = h(y_2)$. By the independence of $h(y)$ and $h(y_2)$, it follows that the probability that $A(h) = h(y) \neq h(y_2)$ is exactly $\frac{1}{4}$. Therefore, the probability that in all the last $\log(\frac{1}{\varepsilon})$ steps it holds that $A(h) = h(y) = h(y_2)$, is at most $(\frac{3}{4})^{\log(\frac{1}{\varepsilon})} < \varepsilon^{0.4}$. We conclude that the overall success probability of M^A is at most $2 \cdot \varepsilon^{1.4}$.

We note that for the above case, it is easy to present an algorithm that inverts f, using black-box access to A, with probability that is very close to ε. Nevertheless, it is possible that one can generalize and strengthen the above argument to preclude any linearly-preserving black-box reduction from interactive hashing to one-way permutation. Such a separation would be quite informative (an easier task would be to rule out any black-box proof that the NOVY protocol is linearly preserving).

Acknowledgments

We are grateful to Moni Naor and Ronen Shaltiel for helpful conversations.

References

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[CCM98] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. pages 493–502, 1998.

[CS06] Claude Crépeau and George Savvides. Optimal reductions between oblivious transfers using interactive hashing. In *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 201–221. Springer, 2006.

[13] As mentioned in Section 5, the hash functions used by the NOVY protocol are not exactly pairwise independent. However, almost the same argument holds for the NOVY hash functions.

[CW79] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, April 1979.

[DHRS04] Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, pages 446–472, 2004.

[Gol01] Oded Goldreich. Randomized methods in computation - lecture notes. 2001.

[HHK+05] Iftach Haitner, Omer Horvitz, Jonathan Katz, Chiu-Yuen Koo, Ruggero Morselli, and Ronen Shaltiel. Reducing complexity assumptions for statistically-hiding commitment. In *Advances in Cryptology – EUROCRYPT 2005*, pages 58–77, 2005. See also preliminary draft of full version, www.wisdom.weizmann.ac.il/~iftachh/papers/SCfromRegularOWF.pdf.

[HHRS07] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols – A tight lower bound on the round complexity of statistically-hiding commitments. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, 2007.

[HL92] Amir Herzberg and Michael Luby. Pubic randomness in cryptography. In *Advances in Cryptology – CRYPTO '92*, volume 740, pages 421–432. Springer, 1992.

[HNO+07] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil Vadhan. Statistically-hiding commitments and statistical zero-knowledge arguments from any one-way function. Unpublished manuscript, November 2007.

[HR07] Iftach Haitner and Omer Reingold. A new interactive hashing theorem. In *Proceedings of the 18th Annual IEEE Conference on Computational Complexity*, 2007.

[Lin03] Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. *JCRYPTOLOGY*, 2003.

[NOV06] Minh-Huyen Nguyen, Shien Jin Ong, and Salil Vadhan. Statistical zero-knowledge arguments for NP from any one-way function. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 3–14, 2006.

[NOVY98] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for NP using any one-way permutation. *Journal of Cryptology*, 11(2):87–108, 1998. Preliminary version in *CRYPTO'92*.

[NV06] Minh-Huyen Nguyen and Salil Vadhan. Zero knowledge with efficient provers. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 287–295. ACM Press, 2006.

[OVY92] R. Ostrovsky, R. Venkatesan, and M. Yung. Secure commitment against A powerful adversary. In *9th Annual Symposium on Theoretical Aspects of Computer Science*, volume 577 of *lncs*, pages 439–448, Cachan, France, 13–15 February 1992. Springer.

[OVY93a] Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Fair games against an all-powerful adversary. *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 155–169, 1993. Preliminary version in *SEQUENCES'91*.

[OVY93b] Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Interactive hashing simplifies zero-knowledge protocol design. In *Advances in Cryptology – EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 267–273. Springer, 1993.

A A Simpler Construction of Statistically Hiding Commitment from Known Regular One-way Functions

In this section we use our new interactive hashing theorem (Theorem 3.8) to give a direct construction of statistically hiding commitment from known regular one-way functions. A commitment scheme is a two-stage protocol between a sender and a receiver. In the first stage, called the ***commit stage***, the sender commits to a private string σ. In the second stage, called the ***reveal stage***, the sender reveals σ and *proves* that it was the value to which she committed in the first stage. We require two properties of commitment schemes. The hiding property says that the receiver learns nothing about σ in the commit stage. The binding property says that after the commit stage, the sender is bound to a particular value of σ; that is, she cannot successfully open the commitment to two different values in the reveal stage. In a statistically hiding and computationally binding commitment scheme. the hiding holds information theoretically (i.e., even an all powerful learns nothing about σ), where the binding property only guaranteed to hold against polynomial-time senders. A known regular one-way function is an efficiently computable function that is hard to invert, and all its images have the same (efficiently computable) number of preimages. See [HHK+05] for the formal definitions of these primitives.

Our construction is achieved by applying the new interactive hashing to the output of the one-way function. The new construction somewhat simplifies the previous construction, and proof, of Haitner et al. [HHK+05], which uses an additional hashing step before applying the NOVY protocol.

Let $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ be an efficiently computable function. Let $\mathcal{G}$ and $\mathcal{H}$ be families of functions mapping strings of length $\ell(n)$ to strings of length one and $m(n)$ respectively, and let $\mathsf{IH} = (\mathsf{S}^I, \mathsf{R}^I)$ be a $\mathcal{H}$-interactive hashing protocol. We define the bit-commitment protocol $\mathsf{Com} = (\mathsf{S}, \mathsf{R})$ as follows.

PROTOCOL A.1

The commitment scheme $\mathsf{Com} = (\mathsf{S}, \mathsf{R})$.

Commit stage.

Common input: 1^n.

S's input: $b \in \{0,1\}$.

1. S_c chooses uniformly at random $x \in \{0,1\}^n$ and sets $y = f(x)$.
2. $(\mathsf{S}_c, \mathsf{R}_c)$ runs $(\mathsf{S}^I(y, 1^n), \mathsf{R}^I(1^n))$, with S_c and R_c acting S^I and R^I respectively.
   Let (h, z) be the output of R^I in this execution.

3. S_c chooses uniformly at random $g \in \mathcal{G}$ and sends g, $c = b \oplus g(y)$ to R.

4. S_c locally outputs x and R_c outputs (h, z, g, c).

Reveal stage.

Common input: 1^n, $b \in \{0,1\}$ and (h, z, g, c).

S's input: x.

1. S_r sends x to R_r.

2. R_r accepts if $h(f(x)) = z$ and $g(f(x)) \oplus b = c$.

Because we will only be able to show that Com is somewhat hiding, we first define what it means to for a scheme to be δ-hiding, for some $\delta \in [0,1]$.

DEFINITION A.2 (weakly hiding against semi-hones receivers)
Commitment scheme $\mathsf{Com} = (\mathsf{S}, \mathsf{R})$ is statistically $\delta(n)$-hiding against semi-hones receivers, if the ensembles $\{\text{view}_{\mathsf{R}}(\mathsf{S}(0), \mathsf{R})(1^n)\}_{n\in\mathbb{N}}$ and $\{\text{view}_{\mathsf{R}}(\mathsf{S}(1), \mathsf{R})(1^n)\}_{n\in\mathbb{N}}$ are of statistical distance at most $\delta(n)$, where $\text{view}_{\mathsf{R}}(\mathsf{S}(b), \mathsf{R})$ denotes the view of R in the commit stage interacting with $\mathsf{S}(b)$.

The following lemma states that for the proper choice of f, $\mathcal{G}$, $\mathcal{H}$ and IH, Protocol A.1 is computationally binding and weakly-hiding bit-commitment scheme.

LEMMA A.3
Let $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ be a regular one-way function with regularity r, let $m(n) = n - \lceil \log(r) \rceil - 4$ and let $\mathcal{G}$ and $\mathcal{H}$ be families of pairwise independent hash functions mapping strings of length $\ell(n)$ to strings of length one and $m(n)$ respectively. [14] Finally, let IH be a $\mathcal{H}$-interactive-hashing protocol, which is binding and hiding, against semi-honest receivers, for $L = Im(f)$ and $W = \{(x, f(x)) \ : \ x \in \{0,1\}^n\}$.

Then, Protocol A.1 instantiated with f, $\mathcal{G}$ and IH, is computationally binding and statistically $\frac{107}{128}$-hiding against semi-honest receivers bit-commitment scheme.

Before proving Lemma A.3, let us first use it for reproving [HHK+05, Theorem 4.4]. [15]

THEOREM A.4
If there exists known regular one-way functions, then there exists statistically hiding and computationally binding commitment schemes.

[14] Note that for large enough value of n we are guaranteed that $m > 0$ since otherwise f could not be one way.

[15] [HHK+05, Theorem 4.4] also holds w.r.t. somewhat more general choices of f. It is easy to verify, however, that the proofs given here can be extended also to these settings.

Proof. Let $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$ be a known regular one-way function with regularity r, let $\mathcal{H}$ be a family of pairwise independent Boolean hash function, let $m = n - \lceil \log(r) \rceil - 4$ and let $\mathcal{H}^m$ be the m-product-family of $\mathcal{H}$. Finally, let IH be the $\mathcal{H}^m$-interactive-hashing protocol resulting by the instantiation of Protocol 3.6 with these choices of m and $\mathcal{H}$. By Corollary 3.14, we have that IH is computationally binding and hiding, against semi-honest receivers, for $L = Im(f)$ and $W = \{(x, f(x)) \ : \ x \in \{0,1\}^n\}$.

Let $\mathcal{G}$ be a family of Boolean pairwise independent hash functions. Lemma A.3 yields that Protocol A.1 instantiated with f, $\mathcal{G}$ and IH, is computationally binding and statistically $\frac{107}{128}$-hiding, against semi-honest receivers, bit commitment scheme. Hence, the existence of a full-fledge computationally binding and statistically hiding commitment scheme follows by [HHK+05, Thm. 5.2 and Thm. 6.1]. □

REMARK A.5

Note that knowing the regularity of f in the above theorem is crucial for the (uniform) instantiation of Protocol 3.6.

Proof. (of Lemma A.3)

Binding. We show that any adversary that breaks the binding of the bit-commitment protocol can be trivially used to break the binding of the underlying interactive hashing protocol IH. Specifically, given an adversary A that breaks the binding of the bit-commitment with non-negligible probability, the following algorithm, M^A, uses A to break the binding of IH. Algorithm M^A acts as A in the interaction with R_I, let (h, z) be the public output of R_I that follows this interaction. Note that the only interaction of A with R is the interaction with R_I, therefore from A's point of view it has just took part in a normal execution of $(\mathsf{S}_c, \mathsf{R}_c)$, let (g, c) be A's message that follows this interaction. By the contradiction assumption, in the reveal stage A outputs with non-negligible probability two elements x_0 and x_1 such that for both $i \in \{0,1\}$ it holds that $h(f(x_i)) = z$ and $g(f(x_i)) \oplus i = c$. In particular, it holds that $h(f(x_0)) = h(f(x_1)) = z$ and $f(x_0) \neq f(x_1)$. Thus, by outputting $(x_0, f(x_0))$ and $(x_1, f(x_1))$, M^A breaks the binding of IH.

Hiding. The view of R_c when interacting with S_c consists of the values of h, $z = h(f(x))$, g and $c = b \oplus g(f(x))$. Note that the only difference between a commitment to one and a commitment to zero is the value of c. We show that w.h.p. the set $\{f(x) \ : \ h(f(x)) = z\}$ is not too small (i.e., contains at least 8 elements) and therefore g "divides" w.h.p. this set into two subsets of similar size. It follows that given the view of R_c, probability that b equals zero and the probability that b equals one are not too far from each other and thus the protocol is weakly hiding.

Let us turn to the formal proof. Let v be a possible view of R_c in the interaction with S_c and h, z, g and c be the values of these variables in v. It follows that for both $b \in \{0,1\}$
$\Pr[\mathrm{view}^{\mathsf{IH}}_{\mathsf{R}}(\mathsf{S}(b), \mathsf{R})(1^n) = v] = \frac{1}{|\mathcal{H} \times \mathcal{G}|} \cdot \Pr_{y \leftarrow Im(f)}[h(y) = z] \cdot \Pr_{y \leftarrow Im(f)}[b \oplus g(y) = c \mid h(y) = z]$.

Therefore,

$$\begin{aligned}
&\Delta(\mathrm{view}^{\mathsf{IH}}_{\mathsf{R}}(\mathsf{S}(0),\mathsf{R})(1^n),\mathrm{view}^{\mathsf{IH}}_{\mathsf{R}}(\mathsf{S}(1),\mathsf{R})(1^n))\\
=\;& \frac{1}{2}\sum_{v}\Big|Pr[\mathrm{view}^{\mathsf{IH}}_{\mathsf{R}}(\mathsf{S}(0),\mathsf{R})(1^n)=v]-Pr[\mathrm{view}^{\mathsf{IH}}_{\mathsf{R}}(\mathsf{S}(1),\mathsf{R})(1^n)=v]\Big|\\
=\;& \frac{1}{2}\cdot\frac{1}{|\mathcal{H}\times\mathcal{G}|}\sum_{h,g,z,c}\Pr_{y\leftarrow Im(f)}[h(y)=z]\cdot\Big|\Pr_{y\leftarrow Im(f)}[0\oplus g(y)=c\mid h(y)=z]\\
-\;& \Pr_{y\leftarrow Im(f)}[1\oplus g(y)=c\mid h(y)=z]\Big|.
\end{aligned}$$

Thus,

$$\begin{aligned}
&\Delta(\mathrm{view}^{\mathsf{IH}}_{\mathsf{R}}(\mathsf{S}(0),\mathsf{R})(1^n),\mathrm{view}^{\mathsf{IH}}_{\mathsf{R}}(\mathsf{S}(1),\mathsf{R})(1^n))\\
=\;& \frac{1}{2}\cdot\frac{1}{|\mathcal{H}\times\mathcal{G}|}\sum_{h,g,z}\Pr_{y\leftarrow Im(f)}[h(y)=z]\cdot 2\cdot\Big|\Pr_{y\leftarrow Im(f)}[g(y)=0\mid h(y)=z]\\
-\;& \Pr_{y\leftarrow Im(f)}[g(y)=1\mid h(y)=z]\Big|\\
=\;& \mathop{\mathsf{Ex}}_{y\leftarrow Im(f),h\leftarrow\mathcal{H},g\leftarrow\mathcal{G}}\left[T\leftarrow Im(f)\cap(h)^{-1}(h(y))\;:\;\frac{\Big||T\cap g^{-1}(0)|-|T\cap g^{-1}(1)|\Big|}{|T|}\right].
\end{aligned}$$

The proof is concluded by the following claim.

CLAIM A.6

$$\Pr_{y\leftarrow Im(f),h\leftarrow\mathcal{H},g\leftarrow\mathcal{G}}=\left[T\leftarrow Im(f)\cap(h)^{-1}(h(y))\;:\;\frac{|T\cap g^{-1}(0)|}{|T|}\notin[\frac{1}{4},\frac{3}{4}]\right]<\frac{43}{64}.$$

Proof. The pairwise independence of $\mathcal{H}$ yields (see Corollary 2.4) that $\Pr[\big|Im(f)\cap(h)^{-1}(h(y))\big|<8]<\frac{1}{4}$. Similarly, the pairwise independence of $\mathcal{G}$ yields that $\Pr[T\leftarrow Im(f)\cap(h)^{-1}(h(y))\;:\;\frac{|T\cap g^{-1}(0)|}{|T|}\notin[\frac{1}{4},\frac{3}{4}]\mid|T|\geq 8]<\frac{9}{16}$. Hence, $\Pr[T\leftarrow Im(f)\cap(h)^{-1}(h(y))\;:\;\frac{|T\cap g^{-1}(0)|}{|T|}\notin[\frac{1}{4},\frac{3}{4}]]\leq\frac{1}{4}+\frac{3}{4}\cdot\frac{9}{16}=\frac{43}{64}$. □

We conclude that $\Delta(\mathrm{view}^{\mathsf{IH}}_{\mathsf{R}}(\mathsf{S}(0),\mathsf{R})(1^n),\mathrm{view}^{\mathsf{IH}}_{\mathsf{R}}(\mathsf{S}(1),\mathsf{R})(1^n))$ is bounded by $\frac{43}{64}\cdot 1+\frac{21}{64}\cdot(\frac{3}{4}-\frac{1}{4})=\frac{107}{128}$.

□